Main Design

Exceptions

Charting.Base

Graphic Engine

Graphic Engine D3D9

Charting

Graphic Engine GDI

Graphic Engine D2D1

SharpDX.DXGI

IDrawing

SharpDX.Direct2D1

SharpDX.Direct3D9

SharpDX

SharpDX dlls

From Ms Padiyab dll

Implement dlls

**Charting**

this dll contain the main classes of the charting control. this dll classes is:

**ActiveElement** : Abstract class for all annotation and other active element.

**Annotation** : Abstract class for all type of annotations. Lines and shapes.

**AnnotationCollection** : the collection of the annotations. this class have some helpful methods.

**AnnotationLine** : Abstract class for all type of line annotations. Vertical Annotation line, Horizontal annotation line.

**AnnotationShape** : Abstract class for all type of shape annotations. Annotation Rectangle and Annotation Ellipse.

**AnnotationEllipse** : The class that implement the ellipse class. contain create initialize and draw methods.

**AnnotationRectangle** : The class that implement the rectangle class. contain create initialize and draw methods.

**HorizontalAnnotationLine** : The class that implement the horizontal annotation line class. contain create initialize and draw methods.

**VerticalAnnotationLine** : The class that implement the vertical annotation line class. contain create initialize and draw methods.

**AnnotationType** : This Enum is contains type of the annotation.

**ArrangmentStyle** : This Enum is contains style of the chartarea drawing on the Chart control.

**Axis** : This class implement the axis class. both axis type are implement in this class, Vertical and horizontal.

**AxisCollection** : The collection of the Axis.

**AxisLabel** : This class implement the label of the axis.

**AxisLabelCollection** : The collection of the axis labels.

**AxisType** : This Enum is contains type of the axis.

**Chart** : This class implement the main panel of the chart control.

**ChartArea** : This class implement the all of management for the create , initialize and drawing component.

**ChartAreaCollection** : The collection of the chart area.

**Grid** : This class implement the grid of the chart area.

**GridLine** : This class implement the grid line of the chart area.

**Series** : This class implement the series component.

**SeriesChartType** : This Enum is contains type of the series chart.

**SeriesCollection** : The collection of the chart series.

**IDrawing**

This interface is the main drawing interface for charting control. this dll have six interface.

**IDevice** : Have this methods and properties

void Initialize(Control control);

void Dispose();

void BeginScene();

void EndScene();

void Clear(Color color);

void SetTransform(float w, float h);

bool IsDisposed { get; }

**IEllipse** : Have this methods and properties

float LineWidth { get; set; }

void Draw(RectangleF rectangle, Color borderColor, Color fillColor, float opacity);

void Draw(RectangleF rectangle, Bitmap bitmap, Color borderColor, float opacity = 1);

**ILine** : Have this methods and properties

void Draw(PointF[] points,Color color,float opacity=1);

void Draw(PointF[] points,Color color, float wScale, float hScale,float opacity=1);

void Draw(DataPointCollection points, float xAxesCenterHeight, float yAxesCenterWidth, float HeightRate, float WidthRate, float xAxesMaxValue,Color color,float opacity=1, bool autoShift = true, bool keepInRightOfYAxis = false);

void Dispose();

float Width { get; set; }

Boolean Antialias { get; set; }

Boolean IsDisposed { get; }

**IRectangle** : Have this methods and properties

float LineWidth { get; set; }

void Draw(RectangleF rectangle,Color borderColor,Color fillColor,float opacity);

void Draw(RectangleF rectangle,Bitmap bitmap,Color borderColor, float opacity = 1);

**ITexture** : Have this methods and properties

void Draw(PointF location);

void Draw(PointF location,float rotateAngle,PointF rotateOrigin,bool isVerticalMirror);

void Dispose();

void LoadImage(Bitmap image);

**ITextWriter** : Have this methods and properties

void Dispose();

void Draw(string Text, Rectangle rectangle, FontDrawFlags fontDrawFlag, Color color,float opacity);

**Graphic Engine**

This dll contain two class. first class is the OSInfo class. this class have a method that get we windows name that control working on it.

other class is the GraphicEngine class. this class is the main class of this dll. and have some important and base methods and properties.

**IsInRunMode** : Get is in design mode or in Run mode.

**AutoDetect** : Auto detect graphic engine type and set the graphicEngineName.

And one create method for each of **Device**, **Texture**, **TextWriter**, Line, **Rectangle** and **Ellipse**. in the create methods first check if in design mode or not. then if in the design mode then create an object from the D3D9 dll. else if in run mode check type of the windows if windows upper than win7 use the D2D1 dll else use the D3D9 dll for create above object.

**Graphic Engine D2D1**

This dll is contain the implement of the **IDrawing** interfaces that used **sharpDX** D2D1 filles.

**Graphic Engine D3D9**

This dll is contain the implement of the **IDrawing** interfaces that used **sharpDX** D3D9 files.

**Why use two strategy for graphic engine?**

When we use sharpdx direct3D9 component to drawing charts component in win7 line drawing with a little noise and don't have good face. On the other hand Direct2D1 don't supported in the win XP. So we use OSInfo class to determine the type of the windows, then use GraphicEngine class AutoDetect to select which dll should be load.

**Why SharpDX?**

The solutions that we can used for drawing fast and beautiful are include :

XNA, MonoGame, MDX, SlimDX and SharpDX.

**XNA**, **MonoGame** and **MDX** be un continue, so create a control based on this solution is a void action.

**SlimDX** is work on the Win7 and above version, so our control can't based on it, because we need use in on the Win XP.

**SharpDX** is a best solution.

**Chart control layering**

GPU

DirectX

SharpDX

Direct2D1

Direct3D9

IDrawing

Chart

Graphic Engine

If need to add new graphic engine to drawing with other solution, implement the all interfaces of the **IDrawing** dll file. And add selection condition to the **GraphicEngine** class.

**How it work?**

when a point add to a series:

--- Run the AddNewItem event handler .

--- Find the chartAreas that have this series.

--- For each chartArea after check the IsOptimizeRefreshing call the chartArea Draw method.

--- In ChartArea's Draw method, drawing all component of the chart.

--- Now control waiting for add a new point to the series.**And what can be do in future?**

-- Update the fill with image for the Ellipse and Rectangle in graphic engine D3D9 solution.

-- Update the fill Pattern in graphic engine D3D9 solution.